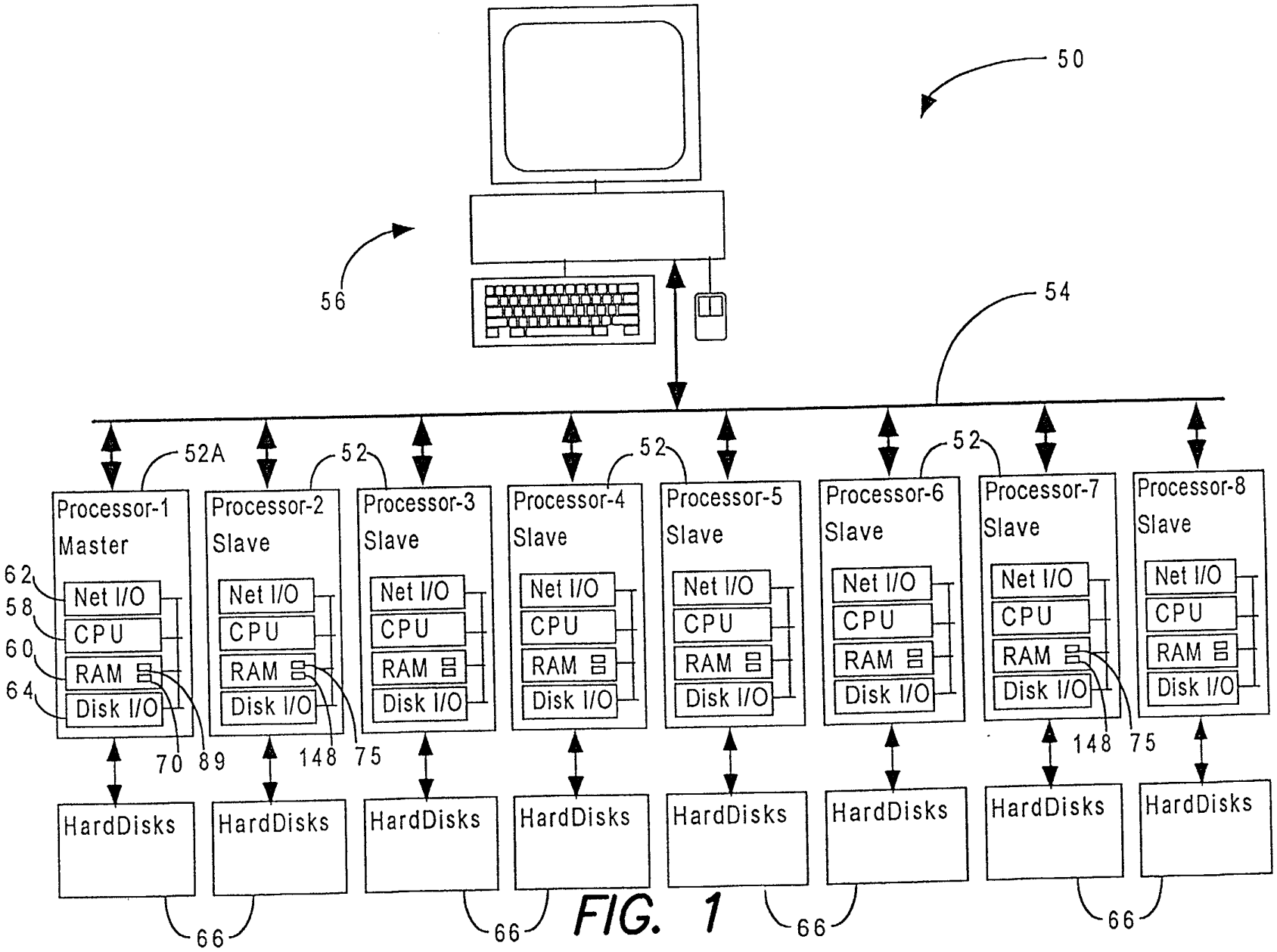
50
56
54
52A
52
52
52
Processor-1
Master
Processor-2
Slave
Processor-3
Slave
Processor-4
Slave
Processor-5
Slave
Processor-6
Slave
Processor-7
Slave
Processor-8
Slave
Net I/O
CPU
RAM
Disk I/O
62
58
60
64
70
89
148
75
148
75
HardDisks
HardDisks
HardDisks
HardDisks
HardDisks
HardDisks
HardDisks
HardDisks
66
66
66
66

FIG. 1

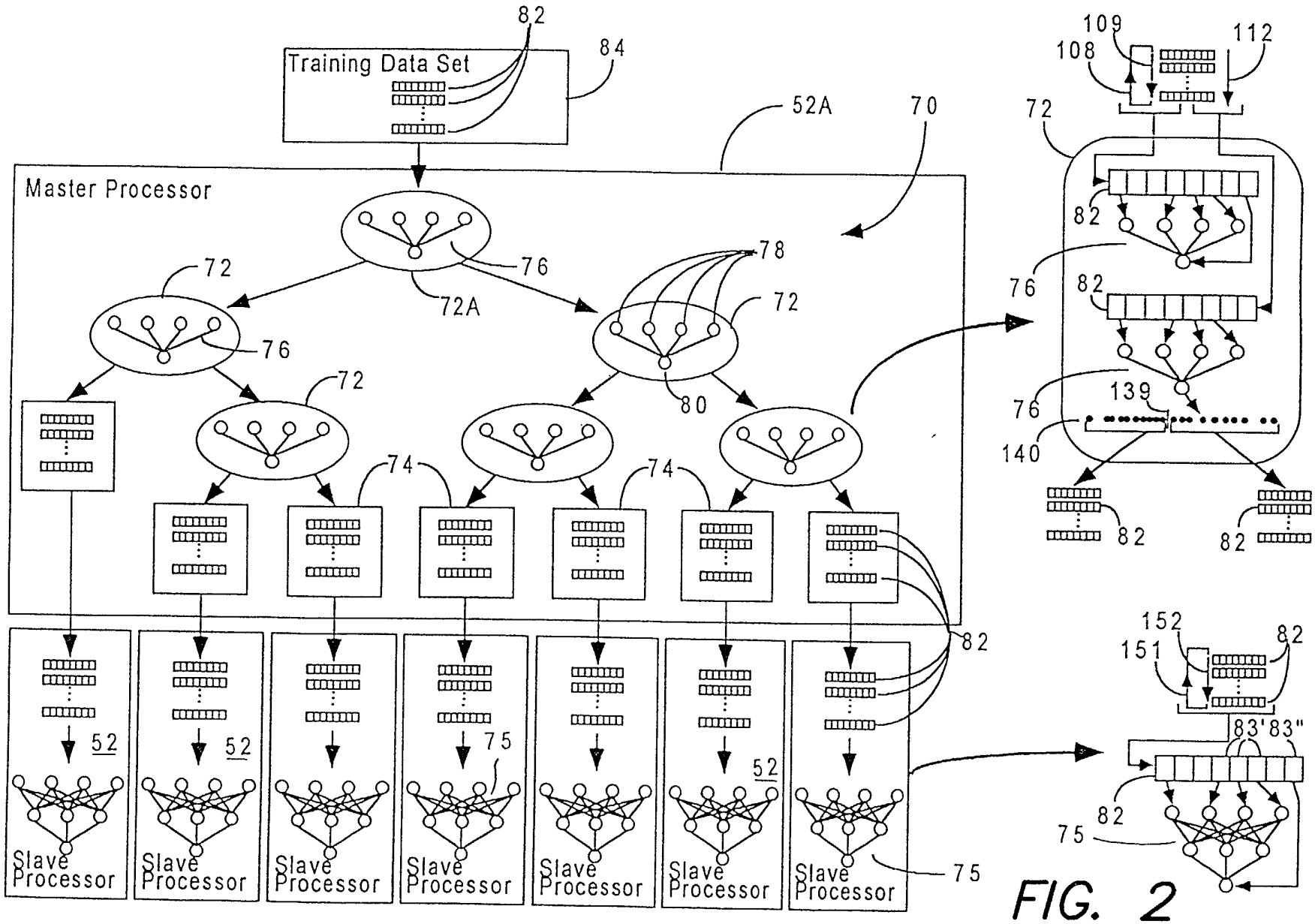

FIG. 2

-BuildModel_Master~89

-make largest full binary tree having NoOfEndNets or less end nodes~90

-add non-terminal nodes to lowest level until have NoOfEndNets end nodes~92

-associate a RecordRatio of 1/NoOfEndNets with each end node~94

-associate with each non-terminal node a RecordRatio equal to the sum of the RecordRatios of its child nodes~96

-supply a set of training records to root node~98

-for each level of tree having non-terminal nodes, starting from the root down~100

-for each non-terminal node in level~102

-from a set of N record parameters, find the ParameterOfGreatest-Spread taken over all record supplied to the tree node~104

-create a two level neural network having an input for each of the N record parameters and having one output~106

-until training converges~108

-for each training record supplied to the tree node~109

-train the weights of the node's network's, appling the record's N parameters to the inputs of the net, and the record's ParameterOfGreatestSpread to its ouput~110

-once the tree node's network has been trained, for each training record passed to the node~112

-re-apply the record's N parameters to the network's inputs~114

-order the record in a ScoreList by the value of net's output~116

-select a SplitPoint such that the ratio of records above and below SplitPoint in ScoreList equals the ratio between the RecordRatios of the node's two child nodes~118

-send all records above SplitPoint to one child node, and all of those below to the other child node~120

-create a compressed representation of the d-tree, including of the weight vector and SplitPoint of each non-terminan node's neural net~122

-for each successive one of the tree's end nodes~124

-cyclically distribute the set of records supplied to that end node to a successive one of NoOfProcessors separate processors~126

-have each of the NoOfProcessor processors execute BuildModel_Slave for each set of end node records destributed to it~129

-once have received compressed representations of the EndNet for each of the d-tree's end nodes, append the compressed representation of each such EndNet to its appropriate place in the compressed d-tree representation, to create a compressed representation of the complete tree network~131

-store compressed complete tree on disk~133

FIG. 3

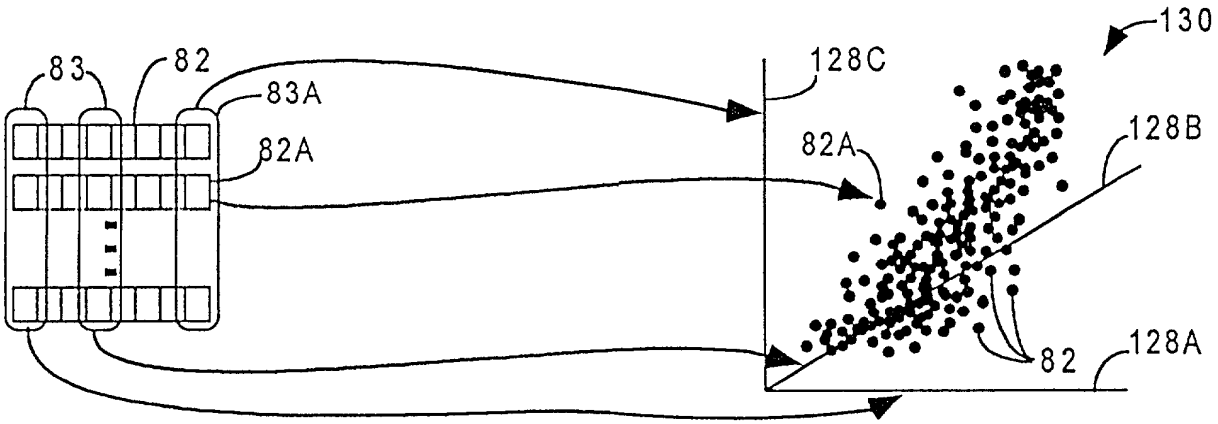

FIG. 4

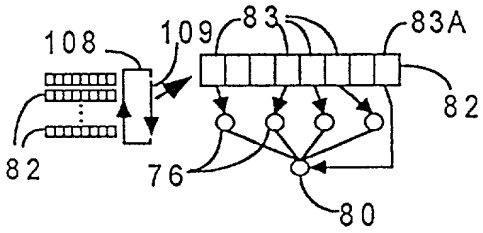

FIG. 5

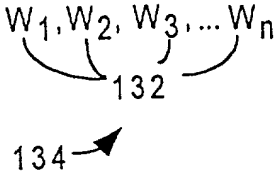

FIG. 6

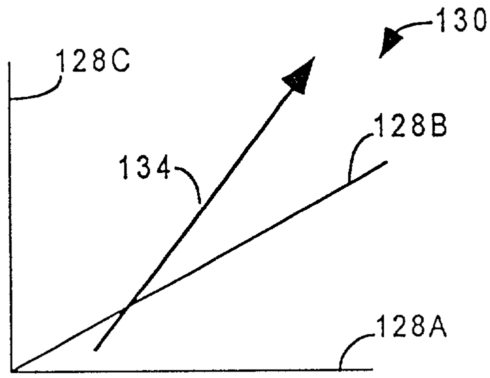

FIG. 7

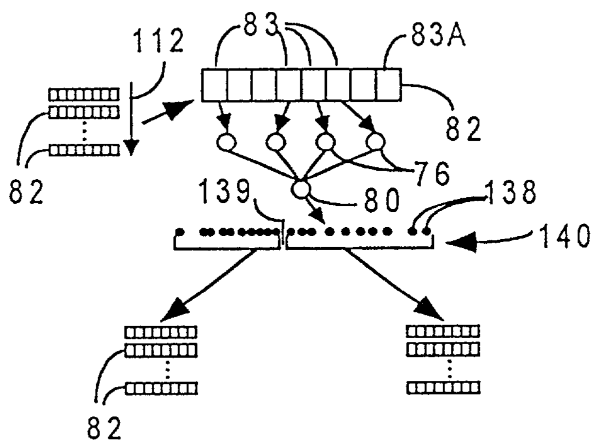

FIG. 8

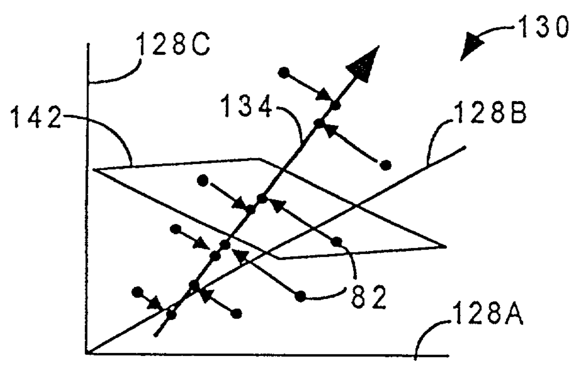

FIG. 9

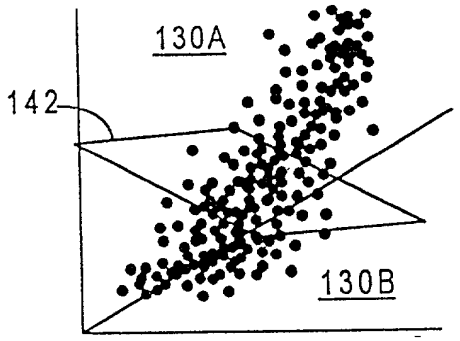

FIG. 10

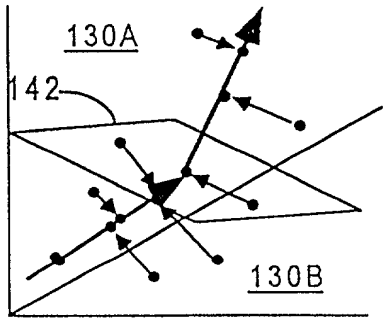

FIG. 11

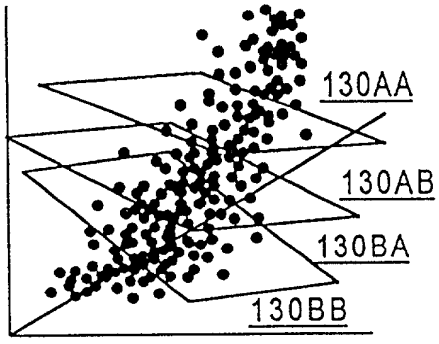

FIG. 12

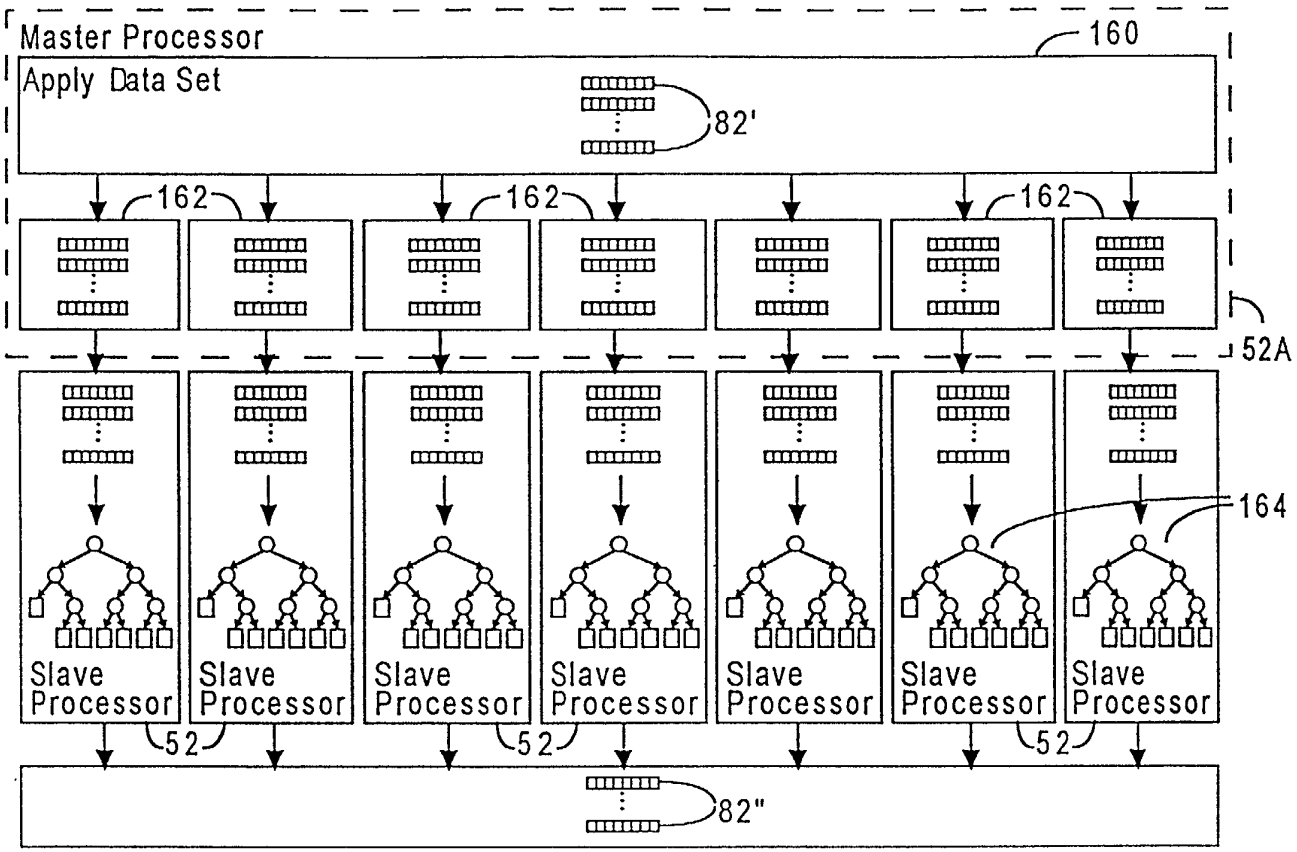

FIG. 14

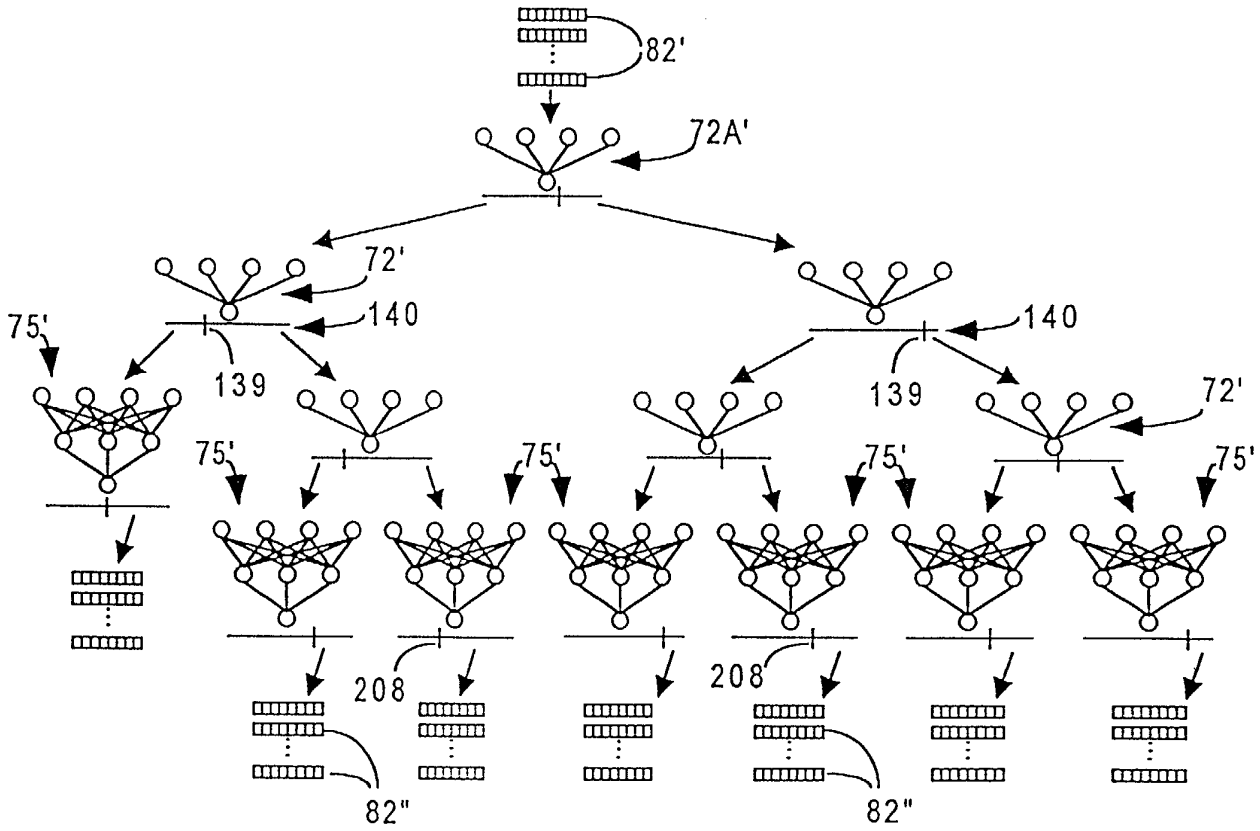

FIG. 15

-BuildModel_Slave~148
-create EndNet, a hidden layer neural network, for the set of end node records having specified list of I input and J output parameters from the set of N parameters used in the d-tree~150
-until training converges~151
-for each record in the end node set~152
-use the record to train the EndNet, appling the values of its I and J parameters, respectively, to the EndNet's input and outputs.~154
-create a compressed representation of the EndNet~156
-send the compressed representation to the Master processor~158

FIG. 13

ApplyModel_Master~170
-if the main data set has not previously been partioned into NoOfProcessors partions, do so~172
-distribute a copy of the compressed complete tree network to each of the NoOfProcessor processors~174
-cause each such processor to run ApplyModel_Slave on its associated partion of the main data set~176
-once receive classification results from such processors, report them~178

FIG. 16

-ApplyModel_Slave~190
-for each record in processor's partion of the main data set~192
-make the root of the compressed complete tree network the CurrentNode~194
-until the CurrentNode is no longer a non-terminal node~196
-multiply record's N parameters by the CurrentNode's associated weight vector to produce a network output value~198
-select one of the CurrentNode's two child nodes the new CurrentNode, depending on whether the node's network output value for the record is greater than or less than the node's SplitPoint~200
-apply the record's I input parameters to the CurrentNode's compressed EndNet~202
-classify the record based on the value of the J outputs of the EndNet, according to specified classification scheme~204
-send specified results of classification to Master~206

FIG. 17

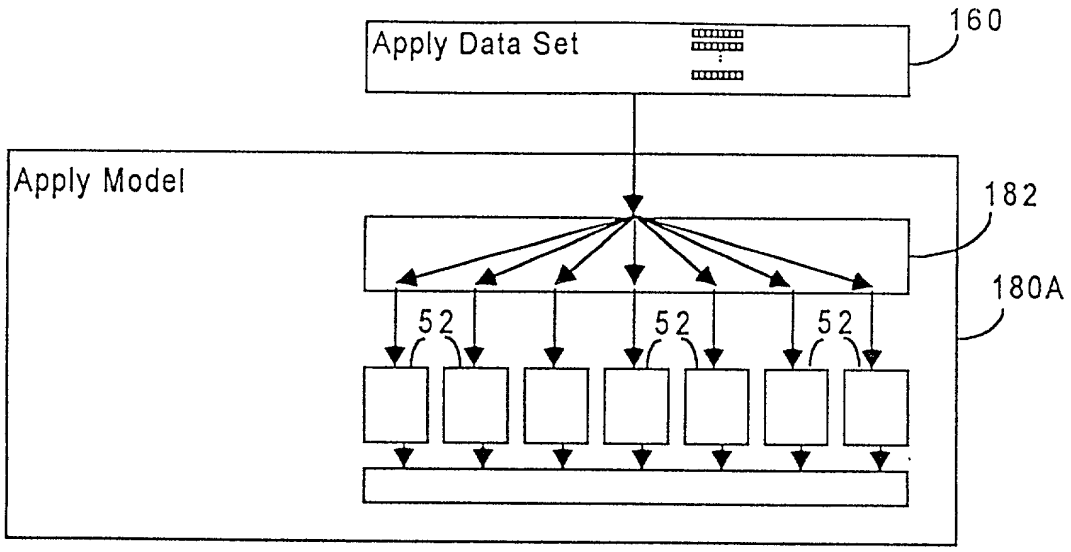
Apply Data Set
160
Apply Model
182
180A
52
52
52

FIG. 18

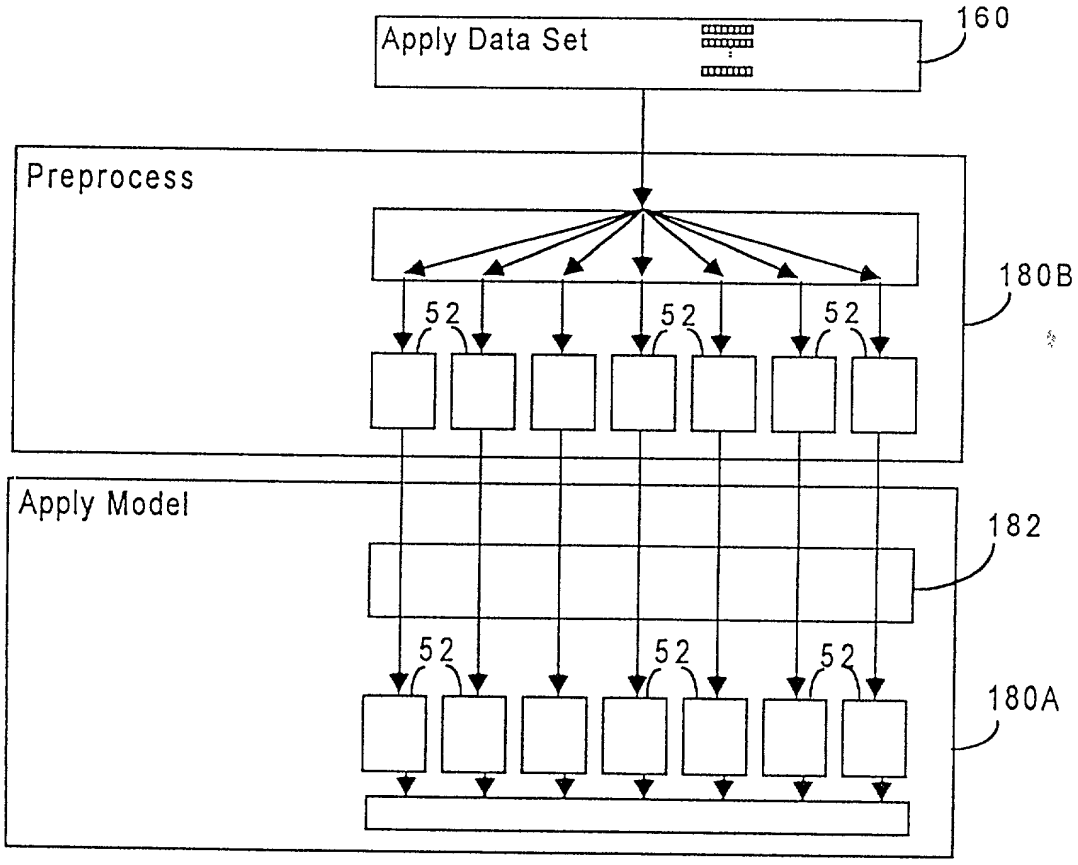
Apply Data Set
160
Preprocess
180B
52
52
52
Apply Model
182
52
52
52
180A

FIG. 19

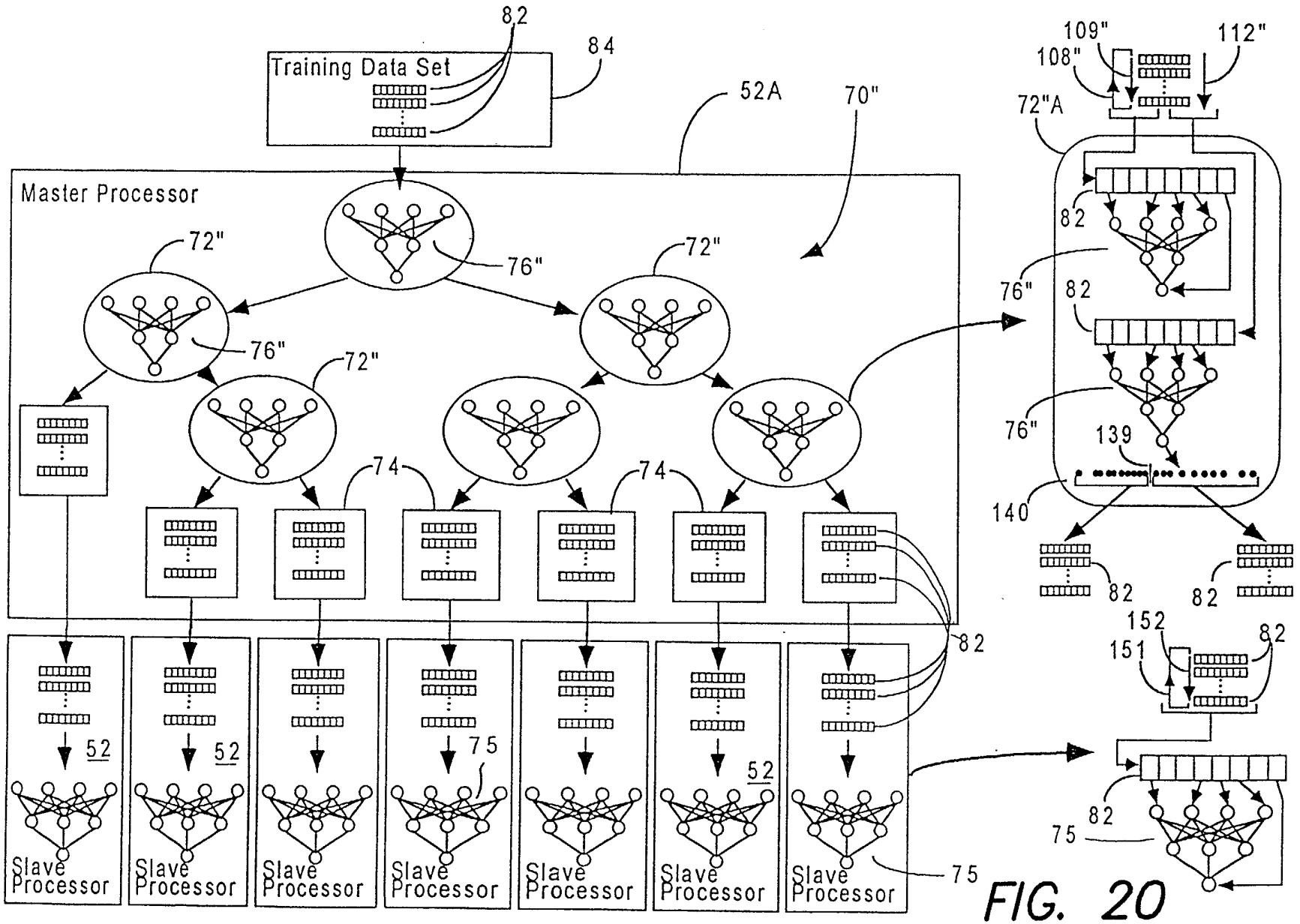
82
Training Data Set
84
52A
70"
Master Processor
72"
76"
72"
76"
72"
74
74
82
52
52
75
52
Slave Processor
Slave Processor
Slave Processor
Slave Processor
Slave Processor
Slave Processor
Slave Processor
75
109"
112"
108"
72"A
82
76"
82
76"
139
140
82
82
152
151
82
82
75

FIG. 20